International Journal of Modern Physics C, Vol. 13, No. 1 (2002) 75–88

A QUANTUM NEURAL NETWORK MODEL

M. ANDRECUT* and M. K. ALI

Department of Physics, University of Lethbridge
4401 University Drive, Lethbridge, Alberta, T1K 3M4, Canada
**mircea.andrecut@uleth.ca*

Received 23 September 2001
Revised 9 October 2001

We present the algorithms necessary for the implementation of a quantum neural network with learning and classification tasks. A complete implementation for the classification and learning algorithms is given in terms of unitary quantum gates. Such a quantum neural network can be used to perform complex classification tasks or to solve the general problem of binary mapping.

Keywords: Quantum algorithms; quantum neural networks.

1. Introduction

In this work we address the classical problems of *learning*, *classification* and *binary mapping*. The task of a classical one-layer perceptron is to classify two classes of patterns by generating a separation hyperplane of the two classes. For linearly separable classes the classical one-layer perceptron is able to construct such a separable hyperplane and all the trained patterns are correctly classified. For nonlinearly separable classes the classical one-layer perceptron algorithm is unable to perform the learning and classification tasks. Also, the one-layer perceptron cannot solve the binary mapping problem. These classical problems can be solved only by using a two-layer perceptron.

We show that, in the case of quantum computing, the concept of separable or nonseparable classes is irrelevant, because a quantum perceptron can learn a superposition of patterns, which are not separable by a hyperplane. The classification mechanism of such a quantum perceptron is assured by a modified version of the Grover algorithm. Also, the binary mapping problem can be solved using only a one-layer quantum perceptron.

2. Some Quantum Computation Concepts

Quantum computation investigates the power of the unique characteristic of quantum systems used as computational machines. A quantum computer is a physical

machine that can process quantum states characterized by a coherent superposition of eigenstates.[1]

In what follows, let us recall a few simple ideas used in quantum computation. A quantum system is described by a quantum state

$$|\psi\rangle = \sum_i c_i |\varphi_i\rangle \,, \tag{1}$$

where $|\varphi_i\rangle$ is a set of quantum eigenstates that form a basis in a Hilbert space. The quantum state $|\psi\rangle$ is said to be in a linear superposition of all the basis states $|\varphi_i\rangle$, and in general case, the coefficients c_i may be complex. This means that in some sense the system is in all basis states at once. However, classically the system can be only in a single basis state.

A quantum system is said to be coherent if it is in a linear superposition of its basis states. If a quantum coherent system interacts in any way with its environment, the superposition is destroyed. This loss of coherence is called decoherence and it is governed by the wave function ψ. The coefficients c_i are called probability amplitudes, and $|c_i|^2 = |\langle\varphi_i|\psi\rangle|^2$ gives the probability of $|\psi\rangle$ collapsing into the state $|\varphi_i\rangle$ if it decoheres. Note that the wave function must collapse to exactly one basis state in a given measurement. It follows that the wave function must satisfy the normalization condition

$$|\langle\psi|\psi\rangle|^2 = \sum_i |c_i|^2 = 1 \,. \tag{2}$$

The simplest spin system, called a spin 1/2 system, whose basis states are represented as $|\uparrow\rangle$ (spin up) and $|\downarrow\rangle$ (spin down), is used as the basic unit in quantum computation. Such a system is referred to as quantum bit or qubit, and renaming the two states $|0\rangle = [1,0]^T$ and $|1\rangle = [0,1]^T$.

The dynamics of the quantum system is described by the temporal Schrödinger equation:

$$\hat{H}|\psi\rangle = i\hbar \frac{\partial}{\partial t} |\psi\rangle \,. \tag{3}$$

The Hamiltonian $\hat{H}$ describes the total energy of the system at a given time and can be very complicated. If the Hamiltonian is time independent, then for an initial value problem with $|\psi(t=0)\rangle = |\psi_0\rangle$, we can define an operator $\hat{U}(t)$ such that

$$\hat{H}\hat{U}(t)|\psi_0\rangle = i\hbar \frac{\partial}{\partial t} \hat{U}(t)|\psi_0\rangle \,, \quad \text{and} \quad \hat{U}(0)|\psi\rangle = |\psi\rangle \,. \tag{4}$$

From here, we get the operator equation

$$\hat{H}\hat{U}(t) = i\hbar \frac{\partial}{\partial t} \hat{U}(t) \tag{5}$$

with the solution

$$\hat{U}(t) = \exp\left(-\frac{i}{\hbar}\hat{H}t\right) = \sum_{n=0}^{\infty} \frac{1}{n!} \left(\frac{-it}{\hbar}\right)^n \hat{H}^n \,. \tag{6}$$

$\hat{U}$ is the operator of temporal evolution and satisfies the criterion

$$\hat{U}(t)|\psi(t=0)\rangle = |\psi(t)\rangle\,. \tag{7}$$

Also, if $|\psi\rangle = \sum_i c_i|\varphi_i\rangle$ is the solution of the time-independent Schrödinger equation

$$\hat{H}|\varphi\rangle = E|\varphi\rangle\,, \tag{8}$$

then

$$|\psi(t)\rangle = \hat{U}(t)|\psi(t=0)\rangle = \sum_i c_i \exp\left(-\frac{i}{\hbar}E_i t|\varphi_i\rangle\right) \tag{9}$$

is the corresponding time dependent solution. Here, $|\varphi_i\rangle$ and E_i are the eigenstates and eigenvalues (energies) of the eigenvalue problem (8).

The operator of temporal evolution satisfies the unitary condition:

$$\hat{U}^\dagger(t)\hat{U}(t) = \exp\left(\frac{i}{\hbar}\hat{H}t\right)\exp\left(-\frac{i}{\hbar}\hat{H}t\right) = 1\,. \tag{10}$$

In general, the operators $\hat{A}$ with $\hat{A}^\dagger = \hat{A}^{-1}$ are called unitary. Here, † stands for complex conjugate transposition of matrices. More generally, unitary opearators can be described as base transformations between two orthonormal bases. Since the temporal evolution of a quantum system is described by a unitary operator and $\hat{U}^\dagger(t) = \hat{U}(-t)$ it follows that the temporal evolution of a quantum system is reversible, as long as no measurement is performed.

The formalism used for quantum computation is based on the concept of quantum gate operations.[2] Such a quantum gate represents a physical implementation of an unitary operator. The quantum gate operations are also reversible, because of the reverse-time evolution specified by the corresponding unitary operators.[3] The quantum gates can be used to implement abstract unitary operations like

$$\mathrm{NOT} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \tag{11}$$

which changes $|0\rangle = [1,0]^T$ to $|1\rangle = [0,1]^T$ (and vice versa), or single qubit rotation defined by the matrix

$$\hat{R}(\theta) = \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix}. \tag{12}$$

It has been shown that a finite set of basic gate operations can be used to construct any quantum gate operation.[4–6] Therefore, the quantum computation consists of a sequence of unitary operators, applied simultaneously to each element of the superposition.

Other two important phenomena exploited in quantum computation and information processing are interference and entanglement.

The interference is explained by the interaction of the wave function with itself through the action of an operator. This way, different parts of the wave function

interfere constructively or destructively according to their relative phases just like any other kind of waves. The interference mechanism represents the basis for the parallel computation in quantum systems. A quantum computer with n physical bits (qubits) can perform unitary operations in a space of 2^n dimensions, exponentially larger than its physical size. Using this massive parallel data processing, a quantum computer can solve some problems, which are believed intractable on classical computers.[7,8]

The entanglement represents the potential for quantum states to exhibit correlations that cannot be accounted for classically states. If we have two independent quantum systems A and B described by the wave functions

$$|\psi^A\rangle = \sum_i a_i |\varphi_i^A\rangle \quad \text{and} \quad |\psi^B\rangle = \sum_i b_i |\varphi_i^B\rangle \,, \tag{13}$$

then the common state is given by the direct product

$$|\psi\rangle = |\psi^A\rangle|\psi^B\rangle = \sum_{i,j} a_i b_j |\varphi_i^A\rangle|\varphi_j^B\rangle \,. \tag{14}$$

Such states are called product states. Unitary operators applied to only one subsystem do not affect the other

$$\hat{U}^A|\psi\rangle = (\hat{U}^A|\psi^A\rangle)|\psi^B\rangle \,. \tag{15}$$

If $|\psi\rangle$ is not a product state, then operations on one subsystem can affect the other and thus we can say that there is a "communication" between them. Two subsystems whose common wave-function is not a product state are entangled.

3. Classical Versus Quantum Perceptron

A classical perceptron[9] is described by the following characteristic function:

$$f(x) = \Theta\left(\sum_{i=1}^{n} w_i x_i + \theta\right) = \begin{cases} 1 & \text{if } \sum_{i=1}^{n} w_i x_i + \theta > 0 \\ 0 & \text{otherwise} \end{cases} , \tag{16}$$

where $x = (x_1, \ldots, x_n)^T$ is the input pattern, $w = (w_1, \ldots, w_n)$ are the synaptic weights and θ is the "threshold". The hyperplane used for the classification task is given by the following equation:

$$\sum_{i=1}^{n} w_i x_i + \theta = 0 \,. \tag{17}$$

The classical learning algorithm consists of modifying the synaptic weights such that the training patterns are correctly classified by the perceptron.[9] However, for nonlinearly separable classes the classical perceptron algorithm is unable to perform the classification task.

An important tentative approach to improve the classical perceptron's performance has been proposed by Gallant.[10] The "pocket" algorithm, developed by

Gallant, is able to construct an optimal hyperplane even in the case of nonseparable classes. The "pocket" algorithm is optimal because it classifies a maximum number of training patterns. In the "pocket" algorithm, the inexact classified patterns lead to a recursive modification of the synaptic vector. Gallant proved that by increasing the number of iterations the synaptic vector approaches to an optimal vector, which define the optimal separation hyperplane. The convergence of the algorithm is only asymptotic and requires a large number of iterations to achieve the optimal synaptic vector. Also, the "pocket" algorithm has been designed only for finite training sets.

Another important tentative method to improve the perceptron performances corresponds to the statistical classification theory.[11] It has been proven the asymptotic equivalence between the following classifiers: neural networks with classification tasks; k-nearest neighbor classifiers; Bayes classifiers.

This result shows that neural networks with classification tasks are equivalent to statistical classifiers. Of course the learning problem can be successfully solved only by using the neural networks, but the classification problem can be optimally solved using only the statistical classifiers.

Now, let us consider the quantum case. Given M distinct binary patterns

$$X = \{|x^i\rangle;\ x^i \in \{0,1\}^n,\ i = 1, \ldots, M,\ M \leq 2^n - 1\}\,, \tag{18}$$

we observe that these patterns can be stored in a single memory element as a superposition of n entangled qubits:

$$|X\rangle = \frac{1}{\sqrt{M}} \sum_{i=1}^{M} |x^i\rangle\,. \tag{19}$$

So far, it is enough to say that the quantum perceptron is characterized by the memory state $|X\rangle$, which can be constructed (learned) using an unitary operator LEARN:

$$|X\rangle \leftarrow \text{LEARN}\,|x^i\rangle\,, \quad i = 1, \ldots, M\,. \tag{20}$$

Later we will show how to generate this state unitarily from a simple initial state of n qubits and we will give a complete implementation of the LEARN operator in terms of elementary quantum gates.

Let us consider also that we have a classification operator CLASSIFY, corresponding to the following characteristic function:

$$\left.\begin{array}{ll} |y\rangle & \text{if } |y\rangle \in X \\ |z\rangle \neq |y\rangle & \text{if } |y\rangle \notin X \end{array}\right\} \leftarrow \text{MEASUREMENT} \leftarrow \text{CLASSIFY}_{|X\rangle}|y\rangle\,. \tag{21}$$

We will show that the CLASSIFY operator can be constructed using a modified version of the Grover algorithm.

We say that the quantum perceptron is a quantum system with n qubits on which one can operate unitarily the LEARN and CLASSIFY algorithms.

One can see that the concept of separable and nonseparable classes is irrelevant from the quantum mechanics point of view, because the quantum perceptron can learn any arbitrary superposition of patterns.

4. The Classification Algorithm

The searching algorithm proposed by Grover is one of the most important successes in quantum computation.[8] Recently, it has been shown that the Grover algorithm can be implemented using the known methods and techniques of quantum optics.[12]

Classically, one needs on average $N/2$ queries to an unsorted database of N items, for finding one item. Grover has shown that, using quantum computation, this problem can be solved using only $\sqrt{N}$ queries.

From the quantum point of view, finding the item in the database means measuring the system and having the system collapse to the eigenstate, which corresponds to the item in the database for which we are searching. The database is composed of all the eigenstates of the quantum system, and it is defined as the uniform superposition of these eigenstates. For a quantum computer with n qubits the database is given by:

$$|\psi\rangle = \frac{1}{\sqrt{N}} \sum_{i=0}^{N-1} |i\rangle \,, \tag{22}$$

where $N = 2^n$ and $|i\rangle$ are the lexicographically ordered eigenstates of the system:

$$|0\rangle = |0 \cdots 00\rangle \,, \quad |1\rangle = |0 \cdots 01\rangle \,, \quad \cdots |N-1\rangle = |1 \cdots 11\rangle \,. \tag{23}$$

In fact, the Grover algorithm is just an algorithm for evolving the wave function $|\psi\rangle$ given by Eq. (22) to a desired eigenstate $|k\rangle$.

The basic idea of the Grover algorithm is to invert the phase of the desired eigenstate using the operator

$$\left.\begin{array}{rl} |i\rangle & \text{if } |i\rangle \neq |k\rangle \\ -|i\rangle & \text{if } |i\rangle = |k\rangle \end{array}\right\} \leftarrow \hat{J}_{|k\rangle}|i\rangle \,, \tag{24}$$

and then to invert all the basis states about the average amplitude of all states using the "diffusion" operator

$$\hat{D} = \hat{I} - 2|\psi\rangle\langle\psi| \,, \tag{25}$$

where $\hat{I}$ is the identity operator.

This sequence of operators is applied a certain number of times

$$T = \frac{\pi\sqrt{N}}{4} \,, \tag{26}$$

and finally one can observe the system:

$$|k\rangle \leftarrow \text{MEASUREMENT} \leftarrow \underbrace{\hat{D}\hat{J}_{|k\rangle} \cdots \hat{D}\hat{J}_{|k\rangle}}_{T \text{ times}} |\psi\rangle \,. \tag{27}$$

The Grover algorithm produces an increase in the amplitude of the desired eigenstate near to unity, followed by a corresponding decrease in the amplitude of the other eigenstates.

The original algorithm can be applied only in the case when all the 2^n eigenstates are in an uniform superposition. Therefore, the Grover algorithm cannot solve the classification problem, where not all the 2^n eigenstates are represented in the database wave function. However, here we give a modified version of the Grover algorithm, which is capable of solving the classification problem.

The main difference between this modified version and the original algorithm is that we assume the database wave function being of the form of the memory element given by Eq. (19). Also, the diffusion operator is restricted to the subspace defined by the patterns stored in the memory:

$$\hat{D}_X = \hat{I} - 2|X\rangle\langle X|\,, \tag{28}$$

where $|X\rangle$ is given by Eq. (19). It can be easily shown that $\hat{D}_X$ is also an unitary operator.

The classification algorithm is then given by:

$$\left.\begin{array}{ll} |y\rangle & \text{if } |y\rangle \in X \\ |z\rangle \neq |y\rangle & \text{if } |y\rangle \notin X \end{array}\right\} \leftarrow \text{MEASUREMENT} \leftarrow \underbrace{\hat{D}_X \hat{J}_{|y\rangle} \cdots \hat{D}_X \hat{J}_{|y\rangle}}_{T \text{ times}} |X\rangle\,. \tag{29}$$

The algorithm evolves the memory state such that if $|y\rangle \in X$, then after a certain number of iterations ($T = \pi\sqrt{M}/4$), the probability to measure $|y\rangle$ is $P = 1$. Also, if $|y\rangle \notin |X\rangle$, then the probability to measure $|y\rangle$ is $P = 0$.

Let us consider the case when $|y\rangle \notin |X\rangle$. Obviously, $\hat{J}_{|y\rangle}|X\rangle = |X\rangle$ and $\hat{D}_X|X\rangle = -|X\rangle$, therefore we have

$$(-1)^T |X\rangle \leftarrow \underbrace{\hat{D}_X \hat{J}_{|y\rangle} \cdots \hat{D}_X \hat{J}_{|y\rangle}}_{T \text{ times}} |X\rangle\,. \tag{30}$$

Thus the probability to measure $|y\rangle$, if $|y\rangle \notin |X\rangle$, is $P = 0$.

Now, let us consider the case when $|y\rangle = |x^k\rangle \in X$. In this case, the memory state $|X\rangle$ can be rewritten as follows

$$|X\rangle = \frac{1}{\sqrt{M}}\,|y\rangle + \sqrt{\frac{M-1}{M}}\,|Z\rangle\,, \tag{31}$$

where

$$|Z\rangle = \frac{1}{\sqrt{M-1}} \sum_{\substack{i=1 \\ i \neq k}}^{M} |x^i\rangle\,. \tag{32}$$

It can be easily shown that the transition element for the modified Grover iteration is given by:

$$\langle y|(\hat{D}_X \hat{J}_{|y\rangle})^t|X\rangle = [1\ \ 0] \begin{bmatrix} \frac{2}{M}-1 & 2\frac{\sqrt{M-1}}{M} \\ -2\frac{\sqrt{M-1}}{M} & \frac{2}{M}-1 \end{bmatrix}^t \begin{bmatrix} \frac{1}{\sqrt{M}} \\ \sqrt{\frac{M-1}{M}} \end{bmatrix}. \tag{33}$$

This result can be rewritten as follows

$$\begin{aligned} \langle y|(\hat{D}_X \hat{J}_{|y\rangle})^t|X\rangle &= [1\ \ 0] \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix}^t \begin{bmatrix} \frac{1}{\sqrt{M}} \\ \sqrt{\frac{M-1}{M}} \end{bmatrix} \\ &= [1\ \ 0] \begin{bmatrix} \cos(t\theta) & \sin(t\theta) \\ -\sin(t\theta) & \cos(t\theta) \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{M}} \\ \sqrt{\frac{M-1}{M}} \end{bmatrix} \\ &= \cos(t\theta-\varphi)\,, \end{aligned} \tag{34}$$

where

$$\begin{aligned} \theta &= \sin^{-1}\left(2\frac{\sqrt{M-1}}{M}\right), \\ \varphi &= \cos^{-1}\left(\frac{1}{\sqrt{M}}\right). \end{aligned} \tag{35}$$

Thus, the probability to measure $|y\rangle$, when $|y\rangle = |x^k\rangle \in X$, is $P = \cos^2(t\theta-\varphi)$. Setting $P = 1$, we obtain

$$T = \frac{\varphi}{\theta} = \frac{\cos^{-1}(1/\sqrt{M})}{\sin^{-1}(2(\sqrt{M-1}/M))}\,. \tag{36}$$

When M is large, we have $\theta \approx 2/\sqrt{M}$ and $\varphi \approx \pi/2$, therefore

$$T = \frac{\pi}{4}\sqrt{M}\,. \tag{37}$$

This way the classification problem is solved.

5. The Learning Algorithm

The learning phase consists in constructing the wave function given by Eq. (19). So far, two algorithms for solving this problem have been proposed.[13,14] The first algorithm[13] requires a polynomial number of elementary operations but also $n+1$ additional qubits. The second algorithm[14] requires also a polynomial number of elementary operations, but without introducing additional qubits. It seems that this

algorithm is more efficient from this point of view. However, we have serious doubts about the formulation of this algorithm because it involves some undetermined operations, like rotations with $0/0$ angles, which requires additional analysis.

Here, we present a different algorithm that requires a polynomial number of elementary operations and $n+1$ additional qubits.

The problem can be formulated as following: the state $|0\cdots0\rangle$ should be transformed, using unitary operations, to a general superposed state having the form given in Eq. (19).

Let us consider that the quantum system is built up from three registers. The first register ($|y\rangle$) contains n qubits and it is used to evolve the state $|\psi\rangle$. The second register contains only one qubit ($|c\rangle$) used to perform some controlled operations. The third register ($|x\rangle$) contains n qubits and it is used to load the given eigenstates (18).

The initial state of the quantum system is given by:

$$|\psi\rangle = |y\rangle|c\rangle|x\rangle = |0\cdots0\rangle|0\rangle|0\cdots0\rangle\,. \tag{38}$$

The algorithm begins with applying the NOT operator to the control qubit:

$$|\psi\rangle = |0\cdots0\rangle|1\rangle|0\cdots0\rangle \leftarrow \mathrm{NOT}_c\,|\psi\rangle\,. \tag{39}$$

The first eigenstate $|x^0\rangle$ is loaded by selectively flipping the qubits of the $|x\rangle$ register:

$$|\psi\rangle = |0\cdots0\rangle|1\rangle|x^0\rangle \leftarrow \mathrm{LOAD}_{|x^0\rangle}\,|\psi\rangle\,. \tag{40}$$

The content of the register $|x\rangle$ is copied into register $|y\rangle$ by applying a CCNOT, or control–control–NOT gate, which performs a NOT on the third qubit (y_k) if and only if the first two (c, x_k) are both in state $|1\rangle$:

$$|\psi\rangle = |x^0\rangle|1\rangle|x^0\rangle \leftarrow \prod_{k=1}^{n} \mathrm{CCNOT}_{(c,x_k),y_k}\,|\psi\rangle\,. \tag{41}$$

The register $|x\rangle$ is transformed to $|1\cdots1\rangle$ using the following operation:

$$|\psi\rangle = |x^0\rangle|1\rangle|1\cdots1\rangle \leftarrow \prod_{k=1}^{n} \mathrm{NOT}_{x_k}\ \mathrm{CCNOT}_{(y_k,c),x_k}\,|\psi\rangle\,. \tag{42}$$

The following n-qubits controlled rotation is applied:

$$\hat{R}^{(1)}_{(x_1,\ldots,x_n),c} = \begin{bmatrix} \sqrt{1-\dfrac{1}{M}} & \dfrac{1}{\sqrt{M}} \\ -\dfrac{1}{\sqrt{M}} & \sqrt{1-\dfrac{1}{M}} \end{bmatrix}. \tag{43}$$

This means that the qubit (c) is rotated if and only if all the control qubits ($x_1,\ldots,x_n$) are in state $|1\rangle$. The result of this operation is:

$$|\psi\rangle = \left(\frac{1}{\sqrt{M}}\,|x^0\rangle|0\rangle + \sqrt{1-\frac{1}{M}}\,|x^0\rangle|1\rangle\right)|1\cdots1\rangle \leftarrow \hat{R}^{(1)}_{(x_1,\ldots,x_n),c}|\psi\rangle\,. \tag{44}$$

We restore $|x\rangle$ to $|0\cdots 0\rangle$:

$$|\psi\rangle = \left(\frac{1}{\sqrt{M}}|x^0\rangle|0\rangle + \sqrt{1-\frac{1}{M}}|x^0\rangle|1\rangle\right)|0\cdots 0\rangle \leftarrow \prod_{k=1}^{n} \mathrm{NOT}_{x_k}|\psi\rangle\,. \tag{45}$$

Now, we copy the content of $|y\rangle$ from the processing term (the term with $|c\rangle = |1\rangle$) into $|x\rangle$:

$$|\psi\rangle = \left(\frac{1}{\sqrt{M}}|x^0\rangle|0\rangle + \sqrt{1-\frac{1}{M}}|x^0\rangle|1\rangle\right)|x^0\rangle \leftarrow \prod_{k=1}^{n} \mathrm{CCNOT}_{(y_k,c),x_k}|\psi\rangle\,. \tag{46}$$

The content of $|y\rangle$ for the processing term is restored to $|0\cdots 0\rangle$:

$$|\psi\rangle = \left(\frac{1}{\sqrt{M}}|x^0\rangle|0\rangle + \sqrt{1-\frac{1}{M}}|0\cdots 0\rangle|1\rangle\right)|x^0\rangle \leftarrow \prod_{k=1}^{n} \mathrm{CCNOT}_{(x_k,c),y_k}|\psi\rangle\,. \tag{47}$$

The second eigenstate $|x^1\rangle$ can be loaded by selectively flipping the qubits of the $|x\rangle$ register:

$$|\psi\rangle = \left(\frac{1}{\sqrt{M}}|x^0\rangle|0\rangle + \sqrt{1-\frac{1}{M}}|0\cdots 0\rangle|1\rangle\right)|x^1\rangle \leftarrow \mathrm{LOAD}_{|x^1\rangle}|\psi\rangle\,. \tag{48}$$

The register $|x\rangle$ is copied into $|y\rangle$ for the processing term:

$$|\psi\rangle = \left(\frac{1}{\sqrt{M}}|x^0\rangle|0\rangle + \sqrt{1-\frac{1}{M}}|x^1\rangle|1\rangle\right)|x^1\rangle \leftarrow \prod_{k=1}^{n} \mathrm{CCNOT}_{(c,x_k),y_k}|\psi\rangle\,. \tag{49}$$

The state of $|x\rangle$ is transformed to $|1\cdots 1\rangle$:

$$|\psi\rangle = \left(\frac{1}{\sqrt{M}}|x^0\rangle|0\rangle + \sqrt{1-\frac{1}{M}}|x^1\rangle|1\rangle\right)|1\cdots 1\rangle$$

$$\leftarrow \prod_{k=1}^{n} \mathrm{NOT}_{x_k}\ \mathrm{CCNOT}_{(y_k,c),x_k}|\psi\rangle\,. \tag{50}$$

The second n-qubits controlled rotation is applied:

$$\hat{R}^{(2)}_{(x_1,\ldots,x_n),c} = \begin{bmatrix} \sqrt{\dfrac{1-2/M}{1-1/M}} & \dfrac{1/\sqrt{M}}{\sqrt{1-1/M}} \\ \dfrac{-1/\sqrt{M}}{\sqrt{1-1/M}} & \sqrt{\dfrac{1-2/M}{1-1/M}} \end{bmatrix}. \tag{51}$$

The result of this operation is:

$$|\psi\rangle = \left(\frac{1}{\sqrt{M}}|x^0\rangle|0\rangle + \frac{1}{\sqrt{M}}|x^1\rangle|0\rangle + \sqrt{1-\frac{2}{M}}|x^1\rangle|1\rangle\right)|1\cdots 1\rangle$$

$$\leftarrow \hat{R}^{(2)}_{(x_1,\ldots,x_n),c}|\psi\rangle\,. \tag{52}$$

We restore $|x\rangle$ to $|0\cdots0\rangle$:

$$|\psi\rangle = \left(\frac{1}{\sqrt{M}}|x^0\rangle|0\rangle + \frac{1}{\sqrt{M}}|x^1\rangle|0\rangle + \sqrt{1-\frac{2}{M}}|x^1\rangle|1\rangle\right)|0\cdots0\rangle$$

$$\leftarrow \prod_{k=1}^{n} \mathrm{NOT}_{x_k}|\psi\rangle\,. \tag{53}$$

We copy the content of $|y\rangle$ from the processing term into $|x\rangle$:

$$|\psi\rangle = \left(\frac{1}{\sqrt{M}}|x^0\rangle|0\rangle + \frac{1}{\sqrt{M}}|x^1\rangle|0\rangle + \sqrt{1-\frac{2}{M}}|x^1\rangle 1\rangle\right)|x^1\rangle$$

$$\leftarrow \prod_{k=1}^{n} \mathrm{CCNOT}_{(y_k,c),x_k}|\psi\rangle\,. \tag{54}$$

The content of $|y\rangle$ for the processing term is restored to $|0\cdots0\rangle$:

$$|\psi\rangle = \left(\frac{1}{\sqrt{M}}|x^0\rangle|0\rangle + \frac{1}{\sqrt{M}}|x^1\rangle|0\rangle + \sqrt{1-\frac{2}{M}}|0\cdots0\rangle|1\rangle\right)|x^1\rangle$$

$$\leftarrow \prod_{k=1}^{n} \mathrm{CCNOT}_{(x_k,c),y_k}|\psi\rangle\,. \tag{55}$$

The next eigenstate can be loaded by selectively flipping the qubits of the $|x\rangle$ register and the same procedure is repeated.

Now, let us consider that we have applied the algorithm for the first $M-1$ eigenstates and the result is given by:

$$|\psi\rangle = \left(\frac{1}{\sqrt{M}}\sum_{i=1}^{M-1}|x^i\rangle|0\rangle + \sqrt{1-\frac{M-1}{M}}|0\cdots0\rangle|1\rangle\right)|x^{M-1}\rangle\,. \tag{56}$$

The next step consists of loading the M eigenstate by selectively flipping the qubits of the $|x\rangle$ register:

$$|\psi\rangle = \left(\frac{1}{\sqrt{M}}\sum_{i=1}^{M-1}|x^i\rangle|0\rangle + \sqrt{1-\frac{M-1}{M}}|0\cdots0\rangle|1\rangle\right)|x^M\rangle \leftarrow \mathrm{LOAD}_{|x^M\rangle}|\psi\rangle\,. \tag{57}$$

The content of $|x\rangle$ is copied into $|y\rangle$ for the processing term:

$$|\psi\rangle = \left(\frac{1}{\sqrt{M}}\sum_{i=1}^{M-1}|x^i\rangle|0\rangle + \sqrt{1-\frac{M-1}{M}}|x^M\rangle|1\rangle\right)|x^M\rangle$$

$$\leftarrow \prod_{k=1}^{n} \mathrm{CCNOT}_{(c,x_k),y_k}|\psi\rangle\,. \tag{58}$$

The state of the $|x\rangle$ register is transformed to $|1\cdots 1\rangle$:

$$|\psi\rangle = \left(\frac{1}{\sqrt{M}}\sum_{i=1}^{M-1}|x^i\rangle|0\rangle + \sqrt{1-\frac{M-1}{M}}\,|x^M\rangle|1\rangle\right)|1\cdots 1\rangle$$

$$\leftarrow \prod_{k=1}^{n}\mathrm{NOT}_{x_k}\,\mathrm{CCNOT}_{(y_k,c),x_k}\,|\psi\rangle\,. \tag{59}$$

The final n-qubits controlled rotation is given by:

$$\hat{R}^{(M)}_{(x_1,\ldots,x_n),c} = \begin{bmatrix} \sqrt{\dfrac{1-M/M}{1-(M-1)/M}} & \dfrac{1/\sqrt{M}}{\sqrt{1-(M-1)/M}} \\ \dfrac{-1/\sqrt{M}}{\sqrt{1-(M-1)/M}} & \sqrt{\dfrac{1-M/M}{1-(M-1)/M}} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}. \tag{60}$$

The result of this operation is:

$$|\psi\rangle = \left(\frac{1}{\sqrt{M}}\sum_{i=1}^{M}|x^i\rangle|0\rangle\right)|1\cdots 1\rangle \leftarrow \hat{R}^{(M)}_{(x_1,\ldots,x_n),c}|\psi\rangle\,. \tag{61}$$

Here, the processing term vanishes and finally, the $|x\rangle$ register can be restored to $|0\cdots 0\rangle$:

$$|\psi\rangle = \left(\frac{1}{\sqrt{M}}\sum_{i=1}^{M}|x^i\rangle\right)|0\rangle|0\cdots 0\rangle \leftarrow \prod_{k=1}^{n}\mathrm{NOT}_{x_k}\,|\psi\rangle\,. \tag{62}$$

One can see that after applying the above described algorithm, the desired state described by Eq. (19) is obtained in the $|y\rangle$ register.

6. The Binary Mapping Problem

Let us consider the general problem of *binary mapping*.[15] Given a set of N input binary patterns

$$X = \{x^i = (x^i_1,\ldots,x^i_n) \in \{0,1\}^n;\ i = 1,\ldots,N\}\,, \tag{63}$$

and a set of N output binary patterns

$$Y = \{y^i = (y^i_1,\ldots,y^i_m) \in \{0,1\}^m;\ i = 1,\ldots,N\}\,, \tag{64}$$

we would like to find a binary mapping function $f\colon X \to Y$ such that

$$f(x^i) = y^i\,, \quad i = 1,\ldots,N\,. \tag{65}$$

The main result regarding the binary mapping problem consists in the following theorem[16]: any single binary pattern is linearly separable from the set of all other binary patterns of same dimension. This theorem has been proven by finding a linear hyperplane separating a binary vertex (pattern) from all other vertices (patterns) in a binary hypercube. The proof of this theorem allows us to build a separation

hyperplane that permits an efficient implementation of the one-layer perceptron. These one-layer perceptrons can be used to synthesize two-layer module for any desired binary mapping.

Classically, a two-layer module for any desired mapping function can be implemented using the one-layer perceptrons as following. The two-layer module has n input, m output and N hidden neurons. For each binary mapping pair (x^k, y^k), $k = 1, \ldots, N$, we consider a hidden neuron (k) with the threshold $\theta^k = \|x^k\|^2 - 1$. The connection weight from the input neuron (i) to this this hidden neuron is $2x_i^k - 1$ and that from this hidden neuron to the output neuron (j) is y_j^k. The threshold for each of the output neurons is set to zero. The activation function for the output neuron is the identity function $(f_0(x) = x)$ and that for the hidden neuron is the hardlimiter function $(f_h(x) = 1$ if $x > 0$, 0 otherwise).

In the quantum case, the binary mapping problem can be solved using only a one-layer composed of m quantum perceptrons. Each perceptron must learn a set of patterns X_k, $k = 1, \ldots, m$, such that the binary mapping described by Eqs. (63)–(65) will be satisfied. These sets of patterns are constructed as follows:

$$|x^i\rangle \in X_k \text{ iff } y_k^i = 1\,, \quad i = 1, \ldots, N\,, \quad k = 1, \ldots, m\,. \tag{66}$$

It is obvious that, if the input pattern for each of these m quantum perceptrons is $|x^i\rangle$ then the measurements, performed after the classification algorithm is applied, will give the following results $|x^i\rangle$ if $y_k^i = 1$ and $|z\rangle \neq |x^i\rangle$ if $y_k^i = 0$. Thus, the binary mapping problem can be solved using only a one-layer composed of m quantum perceptrons.

7. Conclusions

We have presented the algorithms necessary for the implementation of a quantum perceptron. The *classification* problem has been solved using a variant of the Grover searching algorithm. The *learning* algorithm is based mainly on controlled rotations. The general expressions for these rotation operators have been derived recursively. A neural network composed from such quantum perceptrons can be used to perform complex classification tasks. Also, we have shown that the general problem of *binary mapping* can be solved using only a one-layer quantum perceptron.

Acknowledgments

This work has been supported through a grant to M. K. Ali from the Defence Research Establishment Suffield. The authors would also like to thank Dr. S. Barton for his continuing support and helpful discussions.

References

1. A. Steane, *Rep. Progr. Phys.* **61**, 117 (1998).
2. A. Barenco, Ch. Bennet, R. Cleve, D. DiVincenzo, N. Margolus, P. Shor, T. Sleator, J. Smolin, and H. Weinfurter, *Phys. Rev. A* **52**, 3457 (1995).

3. A. Peres, *Phys. Rev. A* **32**, 3266 (1985).
4. D. P. DiVicenzo, *Phys. Rev. A* **51**, 1015 (1995).
5. A. Barenco, *Proc. R. Soc. London*, *Ser. A* **449**, 679 (1995).
6. S. Loyd, *Phys. Rev. Lett.* **75**, 346 (1995).
7. P. Shor, in *Proceedings of the 35th Annual Symposium on the Foundations of Computer Science*, ed. S. Goldwasser (IEEE Computer Society, Los Alamitos, CA, 1994), p. 124.
8. L. K. Grover, *Phys. Rev. Lett.* **79**, 325 (1997).
9. C. M. Bishop, *Int. Neural Networks for Pattern Recognition* (Oxford Univ. Press, 1999).
10. S. Gallant, *Neural Network Learning and Expert Systems* (MIT Press, 1993).
11. D. Ruck, S. Rogers, M. Kabrisky, M. Oxeley, and B. Sutter, *IEEE Trans. Neural Networks* **1**, 296 (1990).
12. M. O. Scully and M. S. Zubairy, *Proc. Natl. Acad. Sci. USA* **98**(17), 9490 (2001).
13. D. Ventura and T. Martinez, *Found. Phys. Lett.* **12**, 547 (1999).
14. G. L. Long and Y. Sun, *Phys. Rev. A* **64**, 014303 (2001).
15. S. Grossberg, *Neural Networks* **1**, 17 (1988).
16. S. C. Huang and Y. F. Huang, *IEEE Trans. Neural Networks* **2**(1), 47 (1991).